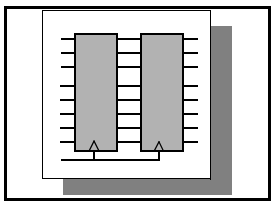
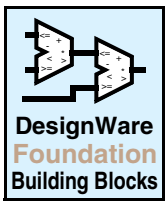




DW03_reg_s_pl

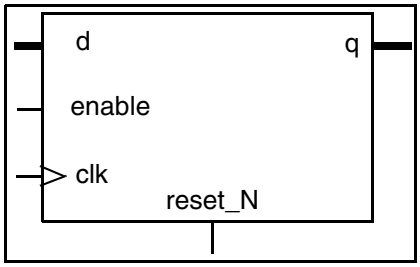
Register with Synchronous Enable Reset

Version, STAR, and myDesignWare Subscriptions: IP Directory

Features and Benefits

- Parameterizable data width
- Parameterized reset to any constant value
- Multiple synthesis implementations

Revision History



Description

DW03_reg_s_pl provides an optimal implementation of a register that is synchronously reset and enabled.

Table 1-1 Pin Description

Pin Name	Width	Direction	Function
d	*width* bits	Input	Input data bus
clk	1 bit	Input	Clock
reset_N	1 bit	Input	Synchronous reset
enable	1 bit	Input	Enables all operations
q	*width* bits	Output	Output data bus

Table 1-2 Parameter Description

Parameter	Values	Description
width	1 to 31 Default: 8	Width of `d` and `q` buses
reset_value	When *width* ≤ 31: 0 to $2^{width} - 1$ When *width* ≥ 32: 0 Default: 0	Resets to a constant

Table 1-3 Synthesis Implementations

Implementation Name	Function	License Feature Required
str	Synthesis model	DesignWare

Table 1-4 Simulation Models

Model	Function
DW03.DW03_REG_S_PL_CFG_SIM	Design unit name for VHDL simulation
dw/dw03/src/DW03_reg_s_pl_sim.vhd	VHDL simulation model source code
dw/sim_ver/DW03_reg_s_pl_sim.v	Verilog simulation model source code

This component is composed of D flip-flops. It can be designed to reset to any constant value. Setup and hold times are relative to the rising edge of the clock signal, `clk`, and are technology dependent.

The `width` parameter configures the width of the part. The `reset_value` parameter indicates the constant value to which you would like the flip-flops set upon reset. This parameter configures logic in front of each internal flip-flop which resets that flip-flop to a one or a zero state corresponding to its bit in the constant `reset_value`.

Due to VHDL and Verilog language limitations, parameters are represented by signed 32-bit integers. The maximum value a parameter can be is 2^{31}–1. This limitation applies to DW03_reg_s_pl's parameters `width` and `reset_value`. If you set `reset_value` to a value that is larger than 0, then you can only use DW03_reg_s_pl up to 31 bits. If you only want to reset DW03_reg_s_pl to 0, then you can use any *width* without limitation.

Suppressing Warning Messages During Verilog Simulation

The Verilog simulation model includes macros that allow you to suppress warning messages during simulation.

To suppress all warning messages for all DWBB components, define the DW_SUPPRESS_WARN macro in either of the following ways:

- Specify the Verilog preprocessing macro in Verilog code:

  ```
  `define DW_SUPPRESS_WARN
  ```

- Or, include a command line option to the simulator, such as:

  `+define+DW_SUPPRESS_WARN` (which is used for the Synopsys VCS simulator)

The warning messages for this model include the following:

- If values other than 1 or 0 are present on a clock port, the following message is displayed:

  ```
  WARNING: <instance_path>.<clock_name>_monitor:
      at time = <timestamp>, Detected unknown value, x, on <clock_name> input.
  ```

  To suppress only this warning message for all DWBB components, use the following macro:

  - Define the DW_DISABLE_CLK_MONITOR macro. You can define this macro in the following ways:
    - Specify the Verilog preprocessing macro in Verilog code:

      ```
      `define DW_DISABLE_CLK_MONITOR
      ```

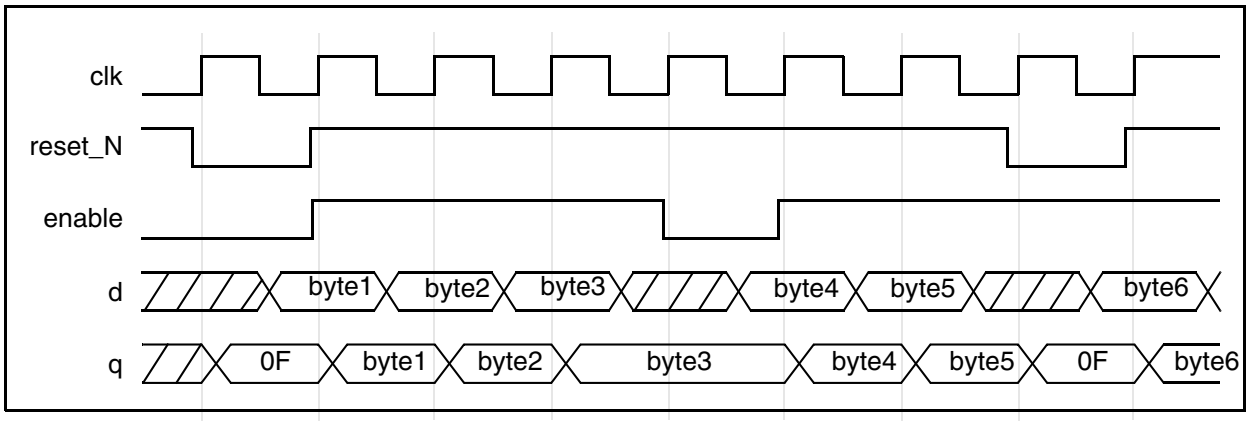
    - Or, include a command line option to the simulator, such as:

      `+define+DW_DISABLE_CLK_MONITOR` (which is used for the Synopsys VCS simulator)

  This message is also suppressed using the DW_SUPPRESS_WARN macro explained earlier.

Timing Diagram

Figure 1-1 Functional Operation - *width* = 8, *reset_value* = 15



Related Topics

- Memory – Registers Overview
- DesignWare Building Block IP User Guide

HDL Usage Through Component Instantiation - VHDL

```
library IEEE,DWARE;
use IEEE.std_logic_1164.all;
use DWARE.DWpackages.all;
use DWARE.DW_foundation_comp.all;

entity DW03_reg_s_pl_inst is
  generic ( inst_width : POSITIVE := 8;
            inst_reset_value : INTEGER := 0 );
  port ( inst_d       : in std_logic_vector(inst_width-1 downto 0);
         inst_clk     : in std_logic;
         inst_reset_N : in std_logic;
         inst_enable  : in std_logic;
         q_inst       : out std_logic_vector(inst_width-1 downto 0) );
end DW03_reg_s_pl_inst;
architecture inst of DW03_reg_s_pl_inst is
begin

  -- Instance of DW03_reg_s_pl
  U1 : DW03_reg_s_pl
    generic map ( width => inst_width,  reset_value => inst_reset_value )
    port map ( d => inst_d,  clk => inst_clk,  reset_N => inst_reset_N,
               enable => inst_enable,  q => q_inst );
end inst;

-- pragma translate_off
configuration DW03_reg_s_pl_inst_cfg_inst of DW03_reg_s_pl_inst is
  for inst
  end for; -- inst
end DW03_reg_s_pl_inst_cfg_inst;
-- pragma translate_on
```

HDL Usage Through Component Instantiation - Verilog

```
module DW03_reg_s_pl_inst( inst_d, inst_clk, inst_reset_N,
                           inst_enable, q_inst );

  parameter width = 8;
  parameter reset_value = 0;

  input [width-1 : 0] inst_d;
  input inst_clk;
  input inst_reset_N;
  input inst_enable;
  output [width-1 : 0] q_inst;

  // Instance of DW03_reg_s_pl
  DW03_reg_s_pl #(width, reset_value)
    U1 ( .d(inst_d), .clk(inst_clk), .reset_N(inst_reset_N),
         .enable(inst_enable), .q(q_inst) );

endmodule
```

Revision History

For notes about this release, see the *DesignWare Building Block IP Release Notes*.

For lists of both known and fixed issues for this component, refer to the STAR report.

For a version of this datasheet with visible change bars, click here.

Date	Release	Updates
July 2020	DWBB_201912.5	■ Adjusted content and title of "Suppressing Warning Messages During Verilog Simulation" on page 3 and added the DW_SUPPRESS_WARN macro
October 2019	DWBB_201903.5	■ Added the "Disabling Clock Monitor Messages" section
April 2019	DWBB_201903.1	■ The 'mbstr' implementation that appeared in Table 1-3 on page 2 has been obsoleted
March 2019	DWBB_201903.0	■ Removed minPower designation from this datasheet
January 2019	DWBB_201806.5	■ Updated example in "HDL Usage Through Component Instantiation - VHDL" on page 4 ■ Added this Revision History table and the document links on this page

Copyright Notice and Proprietary Information

Destination Control Statement

All technical data contained in this publication is subject to the export control laws of the United States of America. Disclosure to nationals of other countries contrary to United States law is prohibited. It is the reader's responsibility to determine the applicable regulations and to comply with them.

Disclaimer

SYNOPSYS, INC., AND ITS LICENSORS MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Trademarks

Synopsys and certain Synopsys product names are trademarks of Synopsys, as set forth at https://www.synopsys.com/company/legal/trademarks-brands.html.

All other product or company names may be trademarks of their respective owners.

Free and Open-Source Software Licensing Notices

If applicable, Free and Open-Source Software (FOSS) licensing notices are available in the product installation.

Third-Party Links

Any links to third-party websites included in this document are for your convenience only. Synopsys does not endorse and is not responsible for such websites and their practices, including privacy practices, availability, and content.

Synopsys, Inc.
www.synopsys.com